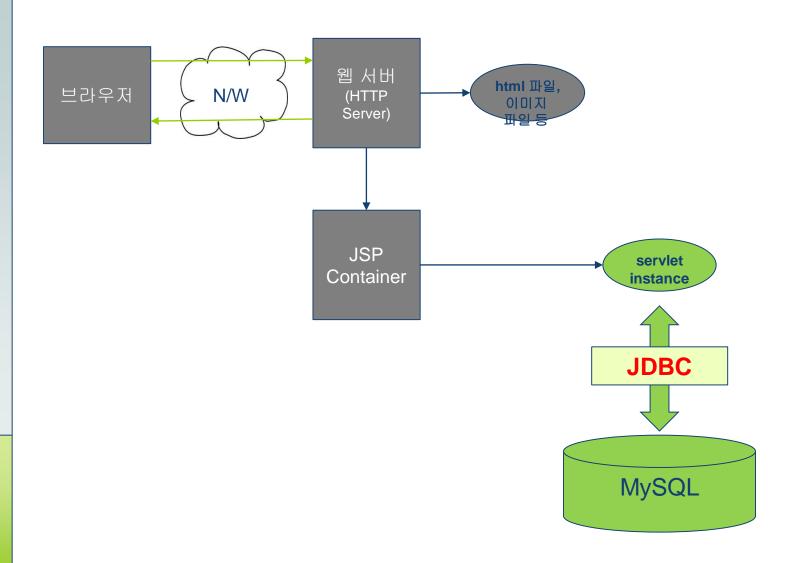
# 》 제10 장 JDBC 프로그래밍

2020년도 1학기

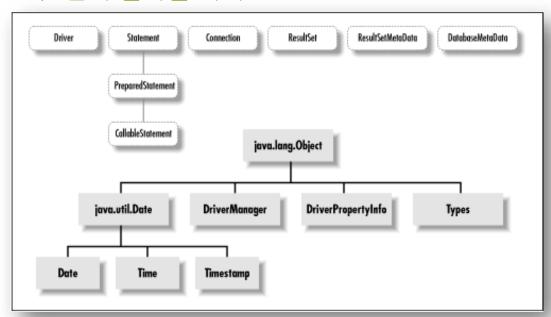
# 테이타베이스 연통



## JDBC OIT

#### JDBC(Java DataBase Connectivity)

- 자바 프로그램에서 데이터베이스와 연결하여 데이터베이스 관련 작업을 할 수 있도록 해주는 자바 프로그래밍 인터페이스를 위한 API(Application Programming Interface) 규격
  - 데이터베이스를 연결하여 테이블 형태의 자료를 참조
  - **SQL** 문을 질의
  - SQL 문의 결과를 처리





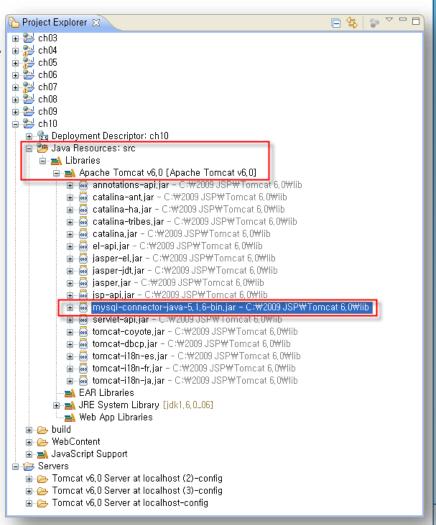
#### ❖JDBC 드라이버

- JDBC 인터페이스에 맞추어 해당 DBMS에서 JDBC 관련 API 호출이 가능하도록 관련 인터페이스와 클래스를 구현한 클래스라이브러리
- JDBC 드라이버를 유형 1, 2, 3, 4로 크게 4가지로 분류
  - JDBC-ODBC 브릿지 드라이버
  - Native-API 드라이버
  - Net-Protocol 드라이버
  - Native-Protocol 드라이버

## MySQL JDBC 三라이버 설치

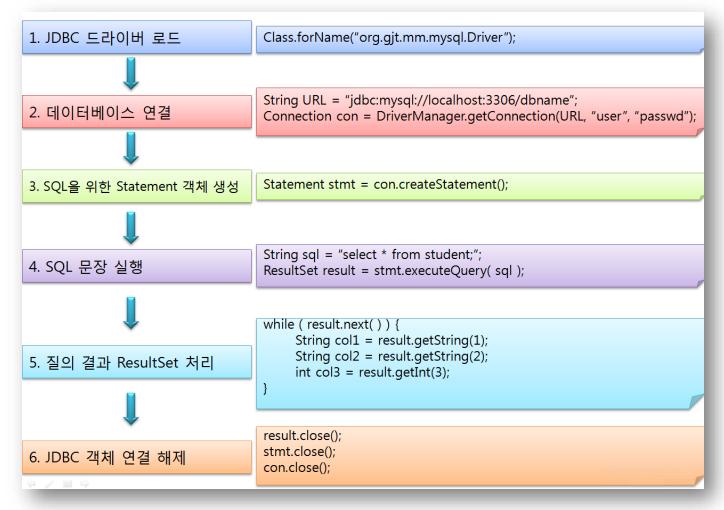
#### ❖JDBC 드라이버

- [mysql-connector-java-5.1.6-bin.jar] 파일을 복사하는
   것으로 완료
  - [JDK 설치 폴더]/[jre]/[lib]/[ext]
  - [Tomcat 설치 폴더]/[lib]
  - [이클립스 프로젝트 하부의 WebContent]/[WEB-INF]/[lib]



#### JDBC 프로그레밍 절차

#### ❖JDBC 프로그래밍 절차 6단계



## 1단계: JDBC 드라이버 로드

#### Class.forName()

```
String driverName = "org.gjt.mm.mysql.Driver";
Class.forName(driverName);
```

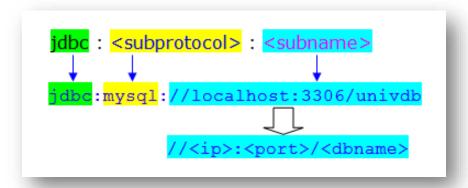
또는

String <u>driverName</u> = "com.mysql.jdbc.Driver"; Class.forName(driverName);

## 2단계: 데이터베이스 연결

### DriverManager.getConnection()

String dbURL = "jdbc:mysql://localhost:3307/chap9ex"; Connection con = DriverManager.getConnection(dbURL, "root", "passwd");



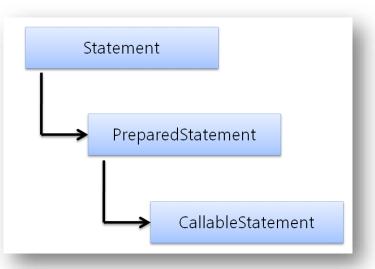
표현 요소	표현 내용	다른 표현	의미
// <host ip="" or=""></host>	//localhost	//203.214.34.67	MySQL이 실행되는 DBMS 서버를 지정, IP주소 또는 도메인 이름
: <port></port>	:3306	:3308	DBMS 서비스 포트 번호로, 3306으로 서비스된다면 생략 가능
/ <dbname></dbname>	/univdb	/mydb	접속할 데이터베이스 이름

# 3단계 : SQL을 위한 Statement 객체 생성

❖질의 문장인 SQL 문을 추상화시킨 인터페이스

객체를 생성

- Statement
- PreparedStatement
- CallableStatement



Statement stmt = con.createStatement();

PreparedStatement pstmt = con.prepareStatement(SQL);

CallableStatement cstmt = con.prepareCall(SQL);

#### 4단계: SQL 문장 실행

- ❖메소드 executeQuery(SQL)
- ❖메소드 executeUpdate(SQL)
- ❖메소드 execute(SQL)

int updateCount = stmt.getUpdateCount();

```
ResultSet result = stmt.executeQuery("select * from student;");
int rowCount = stmt.executeUpdate("delete from student where name = '홍길동';");
stmt.execute("select * from student;");
ResultSet result = stmt.getResultSet();
stmt.execute("delete from student where name = '홍길동';");
```

## 5단계 : 절의 결과 ResultSet 처리

#### ResultSet

ResultSet result = stmt.executeQuery("select \* from student;");

```
while ( result.next() ) {
          <%= result.getString(1) %>
          <%= result.getString("passswd") %>
          <%= result.getString(3) %>
          <%= result.getString("depart") %>
          <%= result.getInt(5) %>
                                           질의 결과인 ResultSet 구조
                                              passwd
                                                                depart
                                                        name
                                        Id
                                                                        year
                                                                                컬럼 이름
                                                                                컬럼 번호
                          질의 결과 처음
                                           BOF(Begin Of File, Before the First Row)
                           커서 위치
                            result, next()
                            로 행을 이동
```

getString("id")

getString(1)

EOF(End OF File, After the Last Row)

getString("name")

getString(3)

현재 커서 위치

현재 커서 행(Row)

getInt("year")

getInt(5)

## 6단계: JDBC 객체 연결 체제

#### ❖JDBC 프로그래밍의 마지막 단계

- 이미 사용한 JDBC 객체의 연결을 해제하는 일
  - con.close();
  - result.close();
  - pstmt.close();

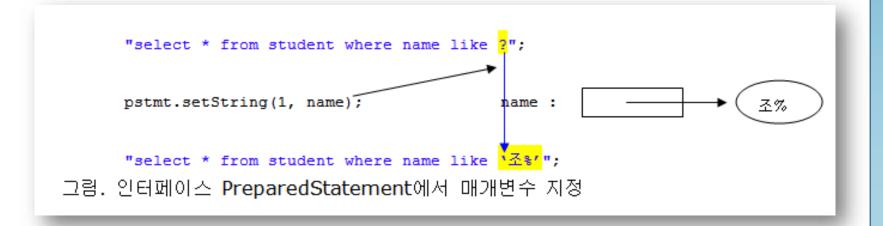
### JSP EIOIEIMOIA 조회 프로그램

- ❖첫 데이터베이스 연결 프로그램
  - dbconnect.jsp
  - 교재 p297
- ❖테이블 조회 프로그램
  - selectdb.jsp
  - 교재 p300

# PreparedStatment를 이용한 검색 프로그램

#### ❖인터페이스 PreparedStatement

- selectname.html
- 교재 p303
- selectname.jsp
- 교재 p305





#### ❖메타데이터

- 데이터를 위한 데이터
- 데이터베이스 자체에 대한 정보 또는 테이블 자체 및 칼럼에 대한 정보
- DatabaseMetaData와 ResultSetMetaData를 제공
- resultsetmetadata.jsp
- 교재 p309

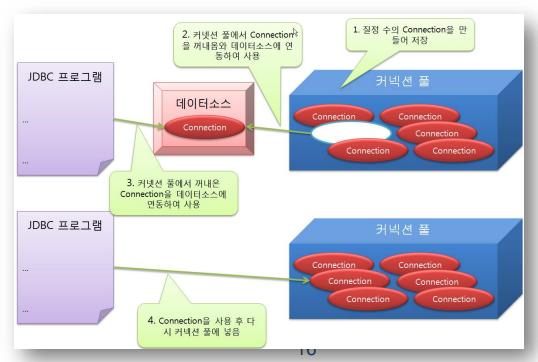
### 커넥션 물

#### ❖ 데이터베이스 커넥션 풀(Database Connection Pool) 관리 기법

- 데이터베이스 연결 작업은 서버의 자원을 이용하는 작업으로, 계속적으로 발생 한다면 시스템에 상당히 부하를 주는 요소
- 일관된 커넥션 관리가 필요

#### ❖ 커넥션 풀(Connection Pool)

미리 여러 개의 데이터베이스 커넥션을 만들어 확보해 놓고 클라이언트의 요청이 있는 경우, 커넥션을 서비스해 주는 커넥션 관리 기법



### 아파치 자카르타 DBCP

#### ❖ 자카르타 DBCP

- 다음의 자카르타 공통 콤포넌트(Jakarta-Commons Component)로 구성
- 톰캣을 설치했다면 이미 자카르타 DBCP가 설치됨
- 톰캣의 설치 폴더 [Tomcat 설치폴더]/[lib] 하부
  - 파일 [tomcat-dbcp.jar]

# 파일 [server.xml]의 컨텍스트에 리소스 설정

- ❖컨텍스트에서 컨넥션 풀에서 이용할 리소스를 추가
  - 데이터베이스의 접속에 필요한 드라이버 이름, 사용자이름, 암호, URL 등을 기술



- ❖ DBCP를 이용한 데이터베이스 연결 프로그램
  - dbconnectwithdbcp.jsp
- ❖ DBCP를 이용한 레코드 삽입 프로그램
  - insertstudent.jsp

